

What is claimed is:

- 1           1.       A system for evaluating a structured message store for message  
2       redundancy, comprising:  
3           a parser extracting a header and a message body from each of a plurality  
4       of messages maintained in a structured message store;  
5           a digester calculating a substantially unique hash code over at least part of  
6       the header and over the message body of each message; and  
7           a message deduper grouping the messages by the hash codes and  
8       identifying one such message as a unique message within each group.
- 1           2.       A system according to Claim 1, further comprising:  
2           a comparer grouping the messages by conversation thread, comparing the  
3       message body for each message within each conversation thread group, and  
4       identifying at least one such message within each conversation thread group as a  
5       unique message.
- 1           3.       A system according to Claim 2, further comprising:  
2           a sorter sorting the messages in each conversation thread group by length,  
3       and identifying one such message having a longest length as the unique message.
- 1           4.       A system according to Claim 1, wherein a plurality of the  
2       messages each further comprise at least one attachment, for each such message,  
3       further comprising:  
4           an attachment digester calculating a substantially unique hash code over at  
5       least part of the at least one attachment for each such message;  
6           a comparer, comprising:  
7           a concatenator logically concatenating the hash code for the  
8       message and the hash code for the at least one attachment; and  
9           an attachment deduper grouping the messages by the logically  
10       concatenated hash codes, comparing the logically concatenated hash codes for  
11       each message within each group, and identifying each message with a matching  
12       logically concatenated hash code as an exact duplicate message.

1           5.       A system according to Claim 4, further comprising:  
2           the comparer identifying each message with one or more hash codes for  
3           the at least one attachment comprising a subset of the hash codes for the at least  
4           one attachment for at least one other message within each group as a near  
5           duplicate message.

1           6.       A system according to Claim 5, further comprising:  
2           the comparer identifying each message without such a subset of the hash  
3           codes as a unique message.

1           7.       A system according to Claim 1, further comprising:  
2           a structured database storing the messages with each message maintained  
3           in a separate indexed record in accordance with a database schema.

1           8.       A system according to Claim 1, further comprising:  
2           a shadow store storing the non-unique messages.

1           9.       A system according to Claim 1, further comprising:  
2           a log tracking message processing.

1           10.      A system according to Claim 1, wherein each message header  
2           further comprises routing data comprising one or more data fields selected from  
3           the group comprising recipient, sender, copy-to, blind-copy-to, and subject.

1           11.      A system according to Claim 1, wherein each hash code is  
2           calculated using a one-way function and comprises alphanumeric, numeric, and  
3           alphabetic character strings.

1           12.      A system according to Claim 11, wherein the one-way function is  
2           selected from the group comprising at least one of the MD5 and secure hashing  
3           algorithms.

1           13.      A method for evaluating a structured message store for message  
2           redundancy, comprising:

3           extracting a header and a message body from each of a plurality of  
4 messages maintained in a structured message store;  
5           calculating a substantially unique hash code over at least part of the header  
6 and over the message body of each message; and  
7           grouping the messages by the hash codes and identifying one such  
8 message as a unique message within each group.

1           14.    A method according to Claim 13, further comprising:  
2           grouping the messages by conversation thread;  
3           comparing the message body for each message within each conversation  
4 thread group; and  
5           identifying at least one such message within each conversation thread  
6 group as a unique message.

1           15.    A method according to Claim 14, further comprising:  
2           sorting the messages in each conversation thread group by length, and  
3 identifying one such message having a longest length as the unique message.

1           16.    A method according to Claim 13, wherein a plurality of the  
2 messages each further comprise at least one attachment, for each such message,  
3 further comprising:  
4           calculating a substantially unique hash code over at least part of the at  
5 least one attachment for each such message;  
6           logically concatenating the hash code for the message and the hash code  
7 for the at least one attachment;  
8           grouping the messages by the logically concatenated hash codes;  
9           comparing the logically concatenated hash codes for each message within  
10 each group; and  
11           identifying each message with a matching logically concatenated hash  
12 code as an exact duplicate message.

1           17.    A method according to Claim 16, further comprising:

2 identifying each message with one or more hash codes for the at least one  
3 attachment comprising a subset of the hash codes for the at least one attachment  
4 for at least one other message within each group as a near duplicate message.

1 18. A method according to Claim 17, further comprising:  
2 identifying each message without such a subset of the hash codes as a  
3 unique message.

1 19. A method according to Claim 13, further comprising:  
2 maintaining a structured database storing the messages with each message  
3 maintained in a separate indexed record in accordance with a database schema.

1 20. A method according to Claim 13, further comprising:  
2 maintaining a shadow store storing the non-unique messages.

1 21. A method according to Claim 13, further comprising:  
2 maintaining a log tracking message processing.

1 22. A method according to Claim 13, wherein each message header  
2 further comprises routing data comprising one or more data fields selected from  
3 the group comprising recipient, sender, copy-to, blind-copy-to, and subject.

1 23. A method according to Claim 13, wherein each hash code is  
2 calculated using a one-way function and comprises alphanumeric, numeric, and  
3 alphabetic character strings.

1 24. A method according to Claim 23, wherein the one-way function is  
2 selected from the group comprising at least one of the MD5 and secure hashing  
3 algorithms.

1 25. A computer-readable storage medium holding code for performing  
2 the method of Claim 13.

1 26. An apparatus for evaluating a structured message store for message  
2 redundancy, comprising:

3 means for extracting a header and a message body from each of a plurality  
4 of messages maintained in a structured message store;  
5 means for calculating a substantially unique hash code over at least part of  
6 the header and over the message body of each message; and  
7 means for grouping the messages by the hash codes and identifying one  
8 such message as a unique message within each group.

1 27. A system for culling duplicative messages maintained in a  
2 structured message store, comprising:  
3 a message extractor retrieving a plurality of messages maintained in a  
4 structured message store with each message comprising a header and a message  
5 body;  
6 a digester calculating a substantially unique hash code over at least part of  
7 the header and over the message body;  
8 a message deduper, comprising:  
9 a comparer comparing the hash codes for each message within  
10 each group; and  
11 a culling module culling each message having an hash code  
12 matching the hash code for at least one other message within the group and  
13 retaining one such non-culled message as a unique message.

1 28. A system according to Claim 27, wherein each such non-culled  
2 message is retained as a potential unique message, further comprising:  
3 a comparer grouping the potential unique messages by conversation thread  
4 and comparing the message body for each potential unique message within each  
5 conversation thread group; and  
6 a culling module culling each potential unique message having a message  
7 body contained within at least one other message within each group and retaining  
8 one such non-culled message as a unique message.

1 29. A system according to Claim 27, further comprising:

2 a sorter sorting the potential unique messages within each group by  
3 conversation thread.

1 30. A system according to Claim 27, wherein a plurality of the  
2 messages each further comprise at least one attachment, further comprising:  
3 the digester calculating a substantially unique hash code over at least part  
4 of the at least one attachment for each message, combining the hash code for each  
5 message and the hash code for the at least one attachment, and grouping the  
6 messages by the combined hash codes;  
7 the comparer comparing the combined hash codes for each message within  
8 each group; and  
9 the culling module culling each message with a matching combined hash  
10 codes and retaining one such non-culled message as a unique message.

1 31. A system according to Claim 30, further comprising:  
2 the comparer identifying each message with one or more hash codes for  
3 the at least one attachment comprising a subset of the hash codes for the at least  
4 one attachment for at least one other message within each group; and  
5 the culling module culling each message with such a subset of the hash  
6 codes and retaining one such non-culled message as a unique message.

1 32. A method for culling duplicative messages maintained in a  
2 structured message store, comprising:  
3 retrieving a plurality of messages maintained in a structured message store  
4 with each message comprising a header and a message body;  
5 calculating a substantially unique hash code over at least part of the header  
6 and over the message body;  
7 comparing the hash codes for each message within each group; and  
8 culling each message having an hash code matching the hash code for at  
9 least one other message within the group; and  
10 retaining one such non-culled message as a unique message.

1           33.     A method according to Claim 32, wherein each such non-culled  
2 message is retained as a potential unique message, further comprising:  
3           grouping the potential unique messages by conversation thread;  
4           comparing the message body for each potential unique message within  
5 each conversation thread group; and  
6           culling each potential unique message having a message body contained  
7 within at least one other message within each group and retaining one such non-  
8 culled message as a unique message.

1           34.     A method according to Claim 32, further comprising:  
2           sorting the potential unique messages within each group by conversation  
3 thread.

1           35.     A method according to Claim 32, wherein a plurality of the  
2 messages each further comprise at least one attachment, further comprising:  
3           calculating a substantially unique hash code over at least part of the at  
4 least one attachment for each message;  
5           combining the hash code for each message and the hash code for the at  
6 least one attachment;  
7           grouping the messages by the combined hash codes;  
8           comparing the combined hash codes for each message within each group;  
9           culling each message with a matching combined hash codes; and  
10          retaining one such non-culled message as a unique message.

1           36.     A method according to Claim 35, further comprising:  
2           identifying each message with one or more hash codes for the at least one  
3 attachment comprising a subset of the hash codes for the at least one attachment  
4 for at least one other message within each group; and  
5           culling each message with such a subset of the hash codes and retaining  
6 one such non-culled message as a unique message.

1           37.     A computer-readable storage medium holding code for performing  
2     the method of Claim 32.

1           38.     An apparatus for culling duplicative messages maintained in a  
2     structured message store, comprising:  
3           means for retrieving a plurality of messages maintained in a structured  
4     message store with each message comprising a header and a message body;  
5           means for calculating a substantially unique hash code over at least part of  
6     the header and over the message body;  
7           means for comparing the hash codes for each message within each group;  
8     and  
9           means for culling each message having an hash code matching the hash  
10    code for at least one other message within the group; and  
11           means for retaining one such non-culled message as a unique message.